Db2 Hofstetter Aufgabe 5

	Modul – 141 – Übung Database Security M141 – Theorie, Übungen und Praktikum	DBSEC - 1

Übungen zu DB-Security

Ziel der Übung: Sie lernen User für verschiedene Zwecke bereitzustellen und mit minimalen Rechten zu versehen. Beachten Sie, dass Sie die Autorisierungen in zwei Datenbanken (**DBBW001** und **DBBW002**) machen müssen. Speichern Sie die jeweiligen GRANT Statements in den vorbereiteten SQL-Scripts, damit Sie diese als Referenz verwenden können.

Die Resultate müssen auch verifiziert werden, d.h. es genügt nicht, nur die Rechte zu setzen, überprüfen Sie ob die Autorisierungen wie gewünscht funktionieren. Sie können dazu z.B. Verschiedene CLP Sessions öffnen oder in Ihrem SQL-Frontend verschiedene Connections (mit den jeweiligen Benutzern) definieren.

Übung 1

Vorbereitete Scripts: **Ueb1_CREATE_User.sh** (UNIX-Script für das Erstellen User und Gruppen)
Ueb1_GRANT_User.sql (SQL-Script für GRANTs)

- Erstellen Sie in Ihrer VM die UNIX Accounts **dbuser10**, **dbuser11** und **dbuser12** (setzen Sie als Passwort jeweils den Namen des entsprechenden Users). Für das Passwort soll kein Ablauf-Datum gesetzt werden.
- Erstellen Sie zudem beiden UNIX Gruppen **dbusrgrp** und **dbadmgrp**.

Diese Benutzer werden lediglich als „Connection-User" verwendet, d.h. es ist nicht vorgesehen, dass sie als UNIX Accounts für das Ausführen von Programmen verwendet werden (User können den CLP somit nicht aufrufen).

- Dem Benutzer **dbuser10** sollen keine weiteren UNIX Gruppen oder Rechte zugewiesen werden.
- Dem Benutzer **dbuser11** soll zusätzlich die UNIX Gruppe ***dbusrgrp*** zugewiesen werden
- Dem Benutzer **dbuser12** soll zusätzlich die UNIX Gruppe ***dbadmgrp*** zugewiesen werden
- Der Gruppe ***dbusrgrp*** soll das Privileg **DATAACCESS** in den Datenbanken **DBBW001** und **DBBW002** zugewiesen werden
- Der Gruppe ***dbadmgrp*** soll das Privileg **DBADM ohne *Datenzugriff*** in den Datenbanken **DBBW001** und **DBBW002** zugewiesen werden

Hinweise: Group-Commands: `groupadd groupmod groupdel`
User-Commands: `useradd usermod userdel chage passwd`

Damit ein Unix-User automatisch Zugriff auf alle DB2 Tools erhält (z.B. Aufruf des CLP's) müssen verschiedene Environment-Variablen und der PATH konfiguriert werden. Dies erfolgt automatisch, wenn Sie z.B. in **.bashrc** die folgenden Zeilen einfügen (muss für die in diesem Auftrag festgelegten User **nicht** gemacht werden):

```
if [ -f /home/db2inst1/sqllib/db2profile ]; then
    . /home/db2inst1/sqllib/db2profile
fi
```

Übung 2

Mit diesen Benutzern soll eine Connection zur Datenbank **DBBW001** erstellt und jeweils das folgende SQL-Statement (SELECT COUNT(*) …) ausgeführt werden, achten Sie darauf, dass die Verbindung zur DB mit dem richtigen User gemacht wird!!.

```
db2   CONNECT TO DBBW001 USER dbuser10
db2   "SELECT  COUNT(*) FROM  BIBLIO.TARTIKEL"
```

Hinweis: Im vorbereiten Verzeichnis finden Sie SQL-Scripts, die jeweils den CONNECT und das SELECT Statement ausführen (z.B. *Ueb2_DBBW001_dbuser10.sql*).

Dokumentieren Sie das Resultat der einzelnen Abfragen (als Resultat ist entweder die Anzahl Datensätze oder die SQL-Fehler-Meldung zu notieren):

mit **dbuser10** Resultat: _____

_____

mit **dbuser11** Resultat: _____

_____

mit **dbuser12** Resultat: _____

_____

Erstellen Sie in der Datenbank **DBBW002** mit dem **Instanz-User** die Tabelle **BIBLIO.TARTIKEL**. Das CREATE Statement finden Sie im Übungs-Verzeichnis (CREATE_TABLE_TARTIKEL.sql). Führen Sie nun das SELECT Statement nochmals für diese Tabelle in der Datenbank **DBBW002** aus.

mit **dbuser10** Resultat: _____

_____

mit **dbuser11** Resultat: _____

_____

mit **dbuser12** Resultat: _____

_____

Der Unterschied zwischen den beiden Datenbanken ist: die Datenbank **DBBW002** wurde beim CREATE mit der Option **RESTRICT** erstellt, d.h. es werden **keine** Autorisierungen automatisch erteilt (ausser für Developer-DB's sollte dies immer angegeben werden). Somit müssen alle Autorisierungen explizit erteilt werden.

Autorisieren Sie nun die Benutzer **dbuser10**, **dbuser11** und **dbuser12** damit Sie das SELECT-Statement in der Datenbank ausführen können. Speichern Sie sich die GRANT's im File *Ueb2_GRANT_User.sql*.

Übung 3

In dieser Übung sollen Sie mit den Benutzern **dbuser10**, **dbuser11** und **dbuser12** zwei Tabellen in der Datenbank **DBBW002** anlegen. D.h. die Verbindung zur DB muss jeweils mit diesen Benutzern aufgebaut werden

Verwenden Sie dazu das bereitgestellte SQL Script ***Ueb3_CREATE_TABLES.sql***. Beachten Sie, dass die beiden Tabellen im Default-Tablespace angelegt und **kein** Schema gesetzt ist, d.h. die Tabellen sollen im Default-Schema der User angelegt werden.

Was stellen Sie fest und was müssen Sie unternehmen (welche GRANT's müssen Sie erteilen), damit die Tabellen tatsächlich mit diesen Benutzern erstellt werden können?

Notieren Sie sich alle GRANT's, die Sie ausführen im File: *Ueb3_GRANT_User.sql*

Achten Sie darauf, dass Sie lediglich die minimal erforderlichen Rechte erteilen!!!

Übung 4

Neben expliziten Autorisierungen für User und Gruppen können auch Autorisierungen für funktionale Rollen erteilt werden. Diese Rollen können bei Bedarf Benutzern oder Gruppen zugewiesen werden.

Erstellen Sie in der Datenbank **DBBW002** eine Rolle **TESTER**. und erteilen Sie dieser Rolle die unten aufgeführten Privilegien auf alle Tabellen, die Sie in der Übung 3 erstellt haben, d.h. die Rolle TESTER muss die folgenden Rechte haben:

Tabelle:	**DBUSER10.TDBS_PERSON**	Rechte:	SELECT, INSERT, UPDATE und DELETE
Tabelle:	**DBUSER11.TDBS_PERSON**	Rechte:	SELECT, INSERT, UPDATE und DELETE
Tabelle:	**DBUSER12.TDBS_PERSON**	Rechte:	SELECT, INSERT, UPDATE und DELETE
Tabelle:	**DBUSER10.TDBS_ABTEILUNG**	Rechte:	SELECT, INSERT, UPDATE und DELETE
Tabelle:	**DBUSER11.TDBS_ABTEILUNG**	Rechte:	SELECT, INSERT, UPDATE und DELETE
Tabelle:	**DBUSER12.TDBS_ABTEILUNG**	Rechte:	SELECT, INSERT, UPDATE und DELETE

Speichern Sie die entsprechenden GRANT Statement in der Datei *Ueb4_GRANT_ROLE.sql*

Erstellen Sie nun die beiden UNIX Benutzer **tester01** und **tester02** und weisen Sie diesen Benutzern die Rolle **TESTER** zu.

Was muss zudem Autorisiert werden, damit neue Test-User Accounts automatisch den Zugriff auf diese Tabellen erhalten und entsprechende SQL-Statement ausführen können?

Die „Tester" sollen lediglich **minimale** Rechte erhalten (nur was sie tatsächlich benötigen) !!!

Übung 1

```
#!/usr/bin/ksh
#
# ACHTUNG: dies ist ein UNIX Script und KEIN SQL-Script !!!!!
#
# Sie koennen in diesem UNIX Script die notwendigen Statements fuer das Erstellen der
# Gruppen und User erfassen, denken Sie jedoch daran, dass SIe dieses Script als root
# User ausfuehren muessen.
#
# Hier die Commands fuer das Erstellen, Modifizieren und Loeschen von User und Groups:
#
# Groups:     groupadd    groupmod   groupdel
# User:       useradd     usermod    userdel     chage    passwd
#

if [[ $(whoami) != "root" ]]; then
   echo "sie muessen die Gruppen und User mit dem root User anlegen!!!"
   exit 16
fi

# Erstellen der Groups
groupadd dbusrgrp   # Befehl zum Erstellen einer Gruppe mit dem Namen "dbusrgrp"
groupadd dbadmgrp   # Befehl zum Erstellen einer Gruppe mit dem Namen "dbadmgrp"



# Erstellen der User
useradd dbuser10    # Befehl zum Erstellen eines Benutzers mit dem Namen "dbuser10"
passwd dbuser10     # Befehl zum Festlegen des Passworts für den Benutzer "dbuser10"
chage -I -1 -m 0 -M 99999 -E -1 dbuser10   # Befehl zur Änderung der Passwortrichtlinien für den Benutzer
"dbuser10"

useradd dbuser11    # Befehl zum Erstellen eines Benutzers mit dem Namen "dbuser11"
passwd dbuser11     # Befehl zum Festlegen des Passworts für den Benutzer "dbuser11"
chage -I -1 -m 0 -M 99999 -E -1 dbuser11   # Befehl zur Änderung der Passwortrichtlinien für den Benutzer
"dbuser11"

useradd dbuser12    # Befehl zum Erstellen eines Benutzers mit dem Namen "dbuser12"
passwd dbuser12     # Befehl zum Festlegen des Passworts für den Benutzer "dbuser12"
chage -I -1 -m 0 -M 99999 -E -1 dbuser12   # Befehl zur Änderung der Passwortrichtlinien für den Benutzer
"dbuser12"
```

```
usermod -a -G dbusrgrp dbuser11   # Befehl zum Hinzufügen des Benutzers "dbuser11" zur Gruppe "dbusrgrp"
usermod -a -G dbadmgrp dbuser12   # Befehl zum Hinzufügen des Benutzers "dbuser12" zur Gruppe
"dbadmgrp"



exit   # Beenden des Skripts
```

```
--
-- Speichern Sie in diesem SQL Script die notwendigen GRANT Statements
--

CONNECT TO DBBW001;
-- Autorisierungen für die Gruppe dbusrgrp
GRANT DATAACCESS ON DATABASE TO GROUP dbusrgrp;

-- Autorisierungen für die Gruppe dbadmgrp
GRANT DBADM WITHOUT DATAACCESS ON DATABASE TO GROUP dbadmgrp;

CONNECT TO DBBW002;
-- Autorisierungen für die Gruppe dbusrgrp
GRANT DATAACCESS ON DATABASE TO GROUP dbusrgrp;

-- Autorisierungen für die Gruppe dbadmgrp
GRANT DBADM WITHOUT DATAACCESS ON DATABASE TO GROUP dbadmgrp;
```

Übung 2

Dokumentieren Sie das Resultat der einzelnen Abfragen (als Resultat ist entweder die Anzahl Datensätze oder die SQL-Fehler-Meldung zu notieren):

mit **dbuser10** Resultat:

```
SQL0551N  The statement failed because the authorization ID does not have the
required authorization or privilege to perform the operation.  Authorization
ID: "DBUSER10".  Operation: "SELECT". Object: "BIBLIO.TARTIKEL".
SQLSTATE=42501
```

mit **dbuser11** Resultat:

```
[db2inst1@localhost ueb05]$ db2 "SELECT COUNT(*) FROM BIBLIO.TARTIKEL"
```

```
1
-----------
        19

  1 record(s) selected.
```

mit **dbuser12** Resultat:

```
[db2inst1@localhost ueb05]$ db2 "SELECT COUNT(*) FROM BIBLIO.TARTIKEL"
SQL0551N  The statement failed because the authorization ID does not have the
required authorization or privilege to perform the operation.  Authorization
ID: "DBUSER12".  Operation: "SELECT". Object: "BIBLIO.TARTIKEL".
SQLSTATE=42501
```

Erstellen Sie in der Datenbank DBBW002 mit dem Instanz-User die Tabelle BIBLIO.TARTIKEL. Das CREATE Statement finden Sie im Übungs-Verzeichnis (CREATE_TABLE_TARTIKEL.sql). Führen Sie nun das SELECT Statement nochmals für diese Tabelle in der Datenbank DBBW002 aus.

mit **dbuser10** Resultat:

```
[db2inst1@localhost ueb05]$ db2 "SELECT COUNT(*) FROM BIBLIO.TARTIKEL"

1
-----------
         0

  1 record(s) selected.
```

mit **dbuser11** Resultat:

```
[db2inst1@localhost ueb05]$ db2 "SELECT COUNT(*) FROM BIBLIO.TARTIKEL"

1
-----------
         0

  1 record(s) selected.
```

mit **dbuser12** Resultat:

```
[db2inst1@localhost ueb05]$ db2 "SELECT COUNT(*) FROM BIBLIO.TARTIKEL"

1
-----------
```

```
          0

  1 record(s) selected.
```

Übung 3

```
--
-- Create Statements fuer Uebung Database Security
--

CREATE TABLE TDBS_PERSON (
    PERSONID    INTEGER GENERATED BY DEFAULT AS IDENTITY (START WITH 1, INCREMENT BY 1, CACHE 20)
NOT NULL,
    NAME        VARCHAR(50) NOT NULL,
    VORNAME     VARCHAR(50) NOT NULL,
    KLASSE      VARCHAR(25) NOT NULL,
    LEHRBETRIEB VARCHAR(50)
)
;

CREATE TABLE TDBS_ABTEILUNG (
    ABTEILUNGID INTEGER GENERATED BY DEFAULT AS IDENTITY (START WITH 1, INCREMENT BY 1, CACHE 20)
NOT NULL,
    NAMEID      VARCHAR(10) NOT NULL,
    BEZEICHNUNG VARCHAR(50) NOT NULL,
    MANAGERIDFS INTEGER
)
;

CREATE UNIQUE INDEX IU03ABTEILUNG_PK
    ON TDBS_ABTEILUNG (ABTEILUNGID)
    PCTFREE 10
    MINPCTUSED 10
    ALLOW REVERSE SCANS
;

ALTER TABLE TDBS_ABTEILUNG
    ADD CONSTRAINT PK_TDBS_ABTEILUNG
    PRIMARY KEY (ABTEILUNGID)
```

```
;

CREATE UNIQUE INDEX IU03PERSON_PK
    ON TDBS_PERSON (PERSONID)
    PCTFREE 10
    MINPCTUSED 10
    ALLOW REVERSE SCANS
;

ALTER TABLE TDBS_PERSON
    ADD CONSTRAINT PK_TDBS_PERSON
    PRIMARY KEY (PERSONID)
;



SELECT * FROM TDBS_PERSON;

SELECT * FROM TDBS_ABTEILUNG;
```

```
-- Autorisierungen für User dbuser10
GRANT CREATETAB, IMPLICIT_SCHEMA ON DATABASE TO USER dbuser10;
GRANT USE OF TABLESPACE USERSPACE1 TO USER dbuser10;

-- Autorisierungen für User dbuser11
GRANT CREATETAB, IMPLICIT_SCHEMA ON DATABASE TO USER dbuser11;
GRANT USE OF TABLESPACE USERSPACE1 TO USER dbuser11;

-- Autorisierungen für User dbuser12
GRANT CREATETAB, IMPLICIT_SCHEMA ON DATABASE TO USER dbuser12;
GRANT USE OF TABLESPACE USERSPACE1 TO USER dbuser12;
```

Übung 4

```
--
-- Speichern Sie in diesem SQL Script die notwendigen GRANT Statements
--

CONNECT TO DBBW002
```

```
-- Erstellen der Rolle "TESTER"
CREATE ROLE TESTER;

-- Autorisierungen für die Rolle "TESTER"
GRANT SELECT, INSERT, UPDATE, DELETE ON TABLE DBUSER10.TDBS_PERSON TO ROLE TESTER;
GRANT SELECT, INSERT, UPDATE, DELETE ON TABLE DBUSER11.TDBS_PERSON TO ROLE TESTER;
GRANT SELECT, INSERT, UPDATE, DELETE ON TABLE DBUSER12.TDBS_PERSON TO ROLE TESTER;

GRANT SELECT, INSERT, UPDATE, DELETE ON TABLE DBUSER10.TDBS_ABTEILUNG TO ROLE TESTER;
GRANT SELECT, INSERT, UPDATE, DELETE ON TABLE DBUSER11.TDBS_ABTEILUNG TO ROLE TESTER;
GRANT SELECT, INSERT, UPDATE, DELETE ON TABLE DBUSER12.TDBS_ABTEILUNG TO ROLE TESTER;

-- Setzen der Rolle "TESTER" für die Benutzer "tester01" und "tester02"
GRANT ROLE TESTER TO USER tester01;
GRANT ROLE TESTER TO USER tester02;

GRANT CONNECT ON DATABASE TO ROLE TESTER;
grant usage on workload SYSDEFAULTUSERWORKLOAD to ROLE TESTER;
GRANT EXECUTE ON PACKAGE NULLID.SQLC2P31 TO ROLE TESTER;
```

Revision #1
Created 10 December 2023 18:10:15 by Manuel Regli
Updated 10 December 2023 18:13:52 by Manuel Regli